Unsteady Flow Analysis Toolkit (UFAT)

David A. Lane*
Computer Sciences Corporation
NASA Ames Research Center
M/S N258-6
Moffett Field, CA 94035-1000

Abstract

The Unsteady Flow Analysis Toolkit (UFAT) was developed to analyze large 3D unsteady Computational Fluid Dynamics (CFD) flow fields. UFAT computes three types of particle traces from the given unsteady flow data – instantaneous streamlines, pathlines, and streaklines. UFAT allows simultaneous visualization of the vector field and the scalar field by assigning color to the particle traces based on the scalar field. Because UFAT does not store the flow data in the memory all at one time, a large number of time steps can be visualized. Three very large 3D unsteady flow data sets were visualized. From the test, it was found that the time it takes to compute the velocity field can be reduced using an approach called 'lazy-evaluation'.

*This work was performed for NASA under Contract NAS 2-12961.

1 Introduction

Time-dependent (unsteady) flow fields are commonly generated in CFD simulations. Although there are several effective tools for visualizing time-independent (steady) flow fields, very few tools exist for unsteady flow visualization. PLOT3D [5] and FAST [2] both provide several effective techniques for visualizing instantaneous flow fields, and they can be used to analyze unsteady flow fields one time step at a time. However, time is not considered in the calculation. PLOT4D [12], Visual3 [10], and VWT [4] are unsteady flow visualization tools, which consider time in the calculation. In [9], a model was proposed to visualize unsteady flow data using extracts. Recently, [11] proposed a distributed architecture for visualizing large-scale unsteady flow data using concurrent processors running across the network. UFAT was developed to compute particle traces from large 3D unsteady flow data. The particle traces can be computed using a large number of time steps from the flow data without requiring the data to be in memory simultaneously. UFAT computes particle traces from unsteady flow data with moving grids.

2 Features

The following sections describe the features in UFAT 1.0.

2.1 Multi-zone/Iblank Grids

UFAT reads PLOT3D grid and solution files. PLOT3D, which is a very popular flow visualization tool in the CFD user community, supports Chimera overset grids [3]. Chimera grids allow complex geometries to be defined using sub-blocks and iblanks. A cell is iblanked when it is not used in the calculation.

2.2 Moving Grids

Unsteady flow data with moving grids are very common in flow analyses. An example data set is the flow field generated from an aircraft with rotating propellers. UFAT provides the capability to compute particle traces in moving grids. The grid is assumed to be fixed in size while the entire grid or a subset of the grid moves in time.

2.3 Particle Traces

UFAT computes three types of particle traces – instantaneous streamlines, pathlines, and streaklines. An instantaneous streamline is a curve which is tangent to the velocity field at an instant in time [7]. A pathline shows the trajectory of a particle released from a given location, sometimes called the seed location. In an instantaneous flow field, pathlines and streamlines are identical. A streakline is a line joining all the particles that have been released from a given seed location. Streaklines can be simulated by releasing particles continuously from the seed locations. In hydrodynamics, streaklines are simulated by releasing hydrogen bubbles rapidly from several fixed locations. Two other common techniques for streakline simulation are smoke injection and dye advection. UFAT simulates streaklines by releasing particles from the specified seed locations at each time step.

2.4 Scalar Fields

Many flow visualization tools render scalar fields by coloring the grid surfaces based on the scalar fields. For example: temperature, pressure or density may be rendered on the surface of the grid. When particle traces are computed, the scalar field is often visualized by coloring the particle traces based on the scalar field. UFAT allows the the particle traces to be colored by their (x, y or z) coordinate, the time at which the particles were released, or a PLOT3D scalar function. When the number of seed locations is large, it may be difficult or impossible to determine the seed position from which a particle was released. UFAT provides the capability to color the particles by their seed number. A number of seeds can be grouped as a rake, and UFAT will color the particles by their rake number.

2.5 Save/Restore

Some unsteady flow data sets are too large to fit in the physical memory of the system all at one time. Often, the scientist is required to analyze a portion of the data at a time. For instantaneous calculation, this is not a problem since the calculation is based on flow data at some instant in time. However, for pathline and streakline calculations, the particle must be traced through all the time steps unless it reaches the grid boundary. The user is often required to keep all time steps of flow data on-line, but there might not be enough disk space. UFAT provides a feature so that the user does not need to store all time steps of flow data on-line during particle calculation. When this feature is specified, UFAT saves the current particles at the end of a run and restores the particles in the next run. Thus, the particle traces can be computed in several runs.

Another advantage of the save/restore feature is that it allows the user to get a preview of the particle traces. Suppose that there are 3,000 time steps in the flow data set, and the user does not want to compute the particle traces using all 3,000 time steps until the user knows where is a good place to position the seeds. Using the save/restore feature, the user can compute particle traces using only 100 time steps first. If the seed locations specified seem to be reasonable, then the particle trace calculation can be resumed.

2.6 Graphics Metafile

The particle traces computed in UFAT are stored in a graphics metafile. This is implemented for two reasons: (1) a large number of time steps of flow data can be visualized without overflowing the memory and (2) the computed particle traces can be animated repeatedly without any re-calculation. A disadvantage is that a graphics metafile must be created, which is not a problem for most flow data sets since the size of the graphics metafile is considerably smaller than the size of the flow data. Because the graphics routines used to output the particle traces are very primitive, UFAT can be easily ported to different systems. Currently, UFAT stores the graphics metafile in ARCGraph[1] format.

3 Implementation Issues

Several implementation issues are discussed in the following sections.

3.1 Data Structures

UFAT was developed using the C programming language. The input grid and solution files are stored in column-major order, a convention of FORTRAN. UFAT stores the data in column-major order also, which allows the data to be stored without re-ordering.

UFAT stores at most two time steps of flow data in memory during execution. As time advances, the flow data from the earlier time step in memory are replaced with the flow data from the most recent time step. This allows UFAT to handle flow data with a large number of time steps.

[1]A graphics library developed at NASA Ames Research Center.

3.2 Particle Integration

UFAT uses the PLOT3D particle tracing library, which was written in FORTRAN. Because the library calculates instantaneous streamlines only, new subroutines were added to the library so that it calculates pathlines and streaklines. The new subroutines consider time in the integration, and interpolation in time and space are performed when necessary. For flow data with moving grids, the input grids are interpolated in time.

3.3 User Parameters

UFAT currently does not provide a Graphical User Interface (GUI); all user parameters are stored in an ASCII text file, which is read by UFAT during initialization. Some of the parameters include the following:

- The type of the grid file (single-zone or multi-zone.)
- The names of the grid and solution files.
- The scalar variable, which is used to assign colors to the particles.
- The physical time between any two consecutive time steps of the flow data. The time steps are assumed to be uniform, i.e. the time steps are sampled at a fixed time increment.

3.4 Lazy Evaluation

In order to compute particle traces, the velocity field is needed. PLOT3D solution files contain momentum and density. Velocity can be obtained by dividing momentum by density. A simple approach would be to compute the velocity at each grid point. However, this can be very time-consuming, especially if the grid has many grid points. An alternative approach would be to compute the velocity in a grid cell only when one or more particles pass through that cell. This approach is suggested in [9] and will be referred to as the 'lazy-evaluation' approach. The velocity at a grid point only needs to be computed once, when any particle first enters the grid cell. Three data sets were used to compare the performance of the two approaches. The data sets are the Harrier Jet, the Delta Wing, and the SOFIA airplane. A description of these data sets is given in the next section. Table 1 shows the performance numbers obtained by calculating velocity at each grid point.

	Pathline		Streakline		Streamline	
Data	Total	Per TS	Total	Per TS	Total	Per TS
Harrier Jet	36 min	20 s	53 min	30 s	86 min	49 s
Delta Wing	26 min	8 s	98 min	29 s	126 min	38 s
SOFIA	24 min	29 s	83 min	100 s	132 min	156 s

Table 1. Performance numbers obtained without 'lazy-evaluation'.

	Pathline		Streakline		Streamline	
Data	Total	Per TS	Total	Per TS	Total	Per TS
Harrier Jet	7 min	4 s	15 min	9 s	47 min	27 s
Delta Wing	8 min	3 s	36 min	11 s	91 min	27 s
SOFIA	5 min	6 s	32 min	38 s	65 min	78 s

Table 2. Improved performance numbers obtained with 'lazy-evaluation'.

As shown in Table 2, the performance increased when the 'lazy evaluation' approach was used to compute the velocity field. Comparing Tables 1 and 2, the performance numbers of the pathline calculations have improved by 3 to 5 times. Similarly, the performance numbers of the streakline calculation have improved by 2 to 3 times. Finally, streamline calculations have improved by almost 2 times. For larger grids, the performance improvement will be even more significant.

4 Results

As mentioned earlier, three data sets were used to test the performance of UFAT. The Harrier Jet consists of 18 grids and has four rotatable nozzles, which provide vertical/short take-off and landing capability [6]. The Delta Wing consists of four grids and has thrust reverser jets in descent [6]. The Stratospheric Observatory For Infrared Astronomy (SOFIA) is a Boeing 747SP with a three meter telescope mounted in the cavity of the airplane [1]. Table 3 shows the number of grid points and time steps used in the three data sets. Each of the data sets is multizone, i.e., consists of several grids and has iblank data. The disk requirements of the three 3D unsteady flow data sets are given in Table 4.

Data Set	# of Grid Points	# of Grids	# of Time Steps Used
Harrier Jet	2.8 million	18	106
Delta Wing	900 thousand	4	205
SOFIA	3.2 million	35	50

Table 3. A description of the Harrier Jet, Delta Wing, and SOFIA.

Data Set	Grid File Size	Solution File Size	Total Size
Harrier Jet	45 MB	56 MB	6.0 GB
Delta Wing	16 MB	20 MB	4.1 GB
SOFIA	53 MB	66 MB	3.4 GB

Table 4. Required disk space of three large unsteady data sets.

Figure 1 depicts the streaklines computed using the Harrier Jet. The streaklines at the 106th time step are shown. The particles are colored by the time of their release from the given seed locations. Light-blue particles are released at the earliest time, and particles that are released the most recently are colored by magenta (see the color bar shown at the bottom of Figure 1). Figure 2 shows the streaklines at the 135th time steps computed using the Delta Wing. Some seeds were positioned near the jet exits, and some were placed near the ground. The swirling near the ground is caused by the interaction of the particles from the jet exits and the particles from the ground. Figure 3 shows SOFIA with the telescope positioned inside the cavity of the plane. Figure 4 shows the streaklines at the 50th time step. The seeds were positioned vertically inside the cavity of the plane where the telescope is positioned.

Acknowledgments

The author would like to thank Jean Clucas, Al Globus, and Tom Woodrow for reviewed drafts of this paper and for their comments. The author also thanks Pieter Buning and Kalpana Chawla for their suggestions for the streakline algorithm.

References

1. Atwood, C. and Van Dalsem, W., Flowfield Simulation about the SOFIA Airborne Observatory, *30th AIAA Aerospace Sciences Meeting and Exhibit,* Reno, Nevada, January 1992, AIAA 92-0656.

2. Bancroft, G., Merritt, F., Plessel, T., Kelaita, P., McCabe, K., and Globus, A., FAST: A Multi-Processed Environment for Visualization of Computa-

tional Fluid Dynamics, in: A. Kaufman, ed., *Proceedings of Visualization '90,* San Francisco, California, October 1990, pp. 14-27.

3. Benek, J., Buning, P., and Steger, J., A 3-D Chimera Grid Embedding Technique, AIAA 85-1523.

4. Bryson, S. and Levit, C., The Virtual Wind Tunnel, *IEEE Computer Graphics & Applications,* Vol. 12, No. 4, July 1992, pp. 25-34.

5. Buning, P.G. and Steger, J.L., Graphics and Flow Visualization in Computational Fluid Dynamics, *7th Computational Fluid Dynamics Conference,* Ohio, Cincinnati, July 1985, AIAA 85-1507.

6. Chawla, K. and Van Dalsem, W., Numerical Simulation of STOL Operations Using Thrust-Vectoring, *AIAA Aircraft Design Systems Meeting,* Hilton Head, South Carolina, August 1992, AIAA 92-4254.

7. Corrie, I., The Fundamental Mechanics of Fluids, McGraw-Hill, New York, 1974.

8. Crawfis, R. and Max, N., Direct Volume Visualization of Three-Dimensional Vector Fields, in: A. Kaufman and L. Lorensen, eds., *1992 Volume Workshop,* Boston, Massachusetts, October 1992, pp. 55-60.

9. Globus, A., A Software Model for Visualization of Time Dependent 3-D Computational Fluid Dynamics Results, *NASA Ames Research Center, NAS Applied Research Technical Report,* RNR 92-031.

10. Haimes, R. and Giles, M., VISUAL3: Interactive Unsteady Unstructured 3D Visualization, *29th AIAA Aerospace Sciences Meeting and Exhibit,* Reno, Nevada, January 1991, AIAA 91-0794.

11. Haimes, R., A New Architecture for Large-Scale Unsteady Scientific Volumetric Visualization, submitted to *Visualization '93.*

12. Smith, M., Van Dalsem, W., Dougherty, F., and Buning, P., Analysis and Visualization of Complex Unsteady Three-Dimensional Flows, *27th AIAA Aerospace Sciences Meeting,* Reno, Nevada, January 1989, AIAA 89-0139.